Image Manipulation with Generative Adversarial Networks From Style Transfer to Procedural Content Generation

Pierre Fernandez
pierre.fernandez@polytechnique.edu

Paul Jacob
paul.jacob@polytechnique.edu

January 24, 2021

Abstract

This report presents our project for the course Object Recognition and Computer Vision. First, we introduce SinGAN's article[7] and reproduce the original results. Then we propose two extensions, for Style Transfer and for Procedural Content Generation. We provide qualitative and quantitative results, and conduct comparative studies with other existing methods for both extensions. Our code (based on the original implementation) [1].

1. Introduction

Generative Adversarial Networks [2] are an approach for using neural networks as data generators, which have attracted a lot of attention in computer vision thanks to their ability to generate extremely realistic samples. However, GANs present some limitations such as the need of a large training dataset and a lack of control on the generation.

Shaham et al. tackle those issues by proposing SinGAN [7], a generative model which is able to learn internal statistics from a single image. The proposed model contains a pyramid of convolutional GANs, each responsible for learning the patch distribution at a different scale of the image. This allows to generate new samples of arbitrary size, that have significant variability, yet maintain both the global structure and the fine textures of the training image.

In this project we first successfully reproduce SinGAN results on various image manipulation tasks (see figure S1). Then, we show how SinGAN can be used for Style Transfer, by training a SinGAN on a style image and injecting a content image at different scales of the SinGAN's pyramid. We evaluate the method qualitatively on portraits and landscapes, and provide a quantitative analysis of the results with the SIFID metric [6]. A Procedural Content Generation (PCG) goal can be achieved in a similar fashion by training on a well-chosen image level and by generating a semantic level from the image. We provide several qualitative results, as well as quantitative evaluations using the Tile-Pattern KL divergence [5].

[1] https://github.com/Poljy/Recvis_Project

2. Application of SinGAN on Style Transfer

2.1. Related Work

Style Transfer refers to techniques that intend to pick the style of a particular image, and to restore it to another image while preserving its content. The two images are called respectively the *style* image, and the *content* image. One motivation for Style Transfer is purely for creative purposes, by creating artificial artworks of high perceptual quality. Also, it can be used to highlight the capacities and internal representations of neural networks. Finally, it may be useful for the enhancement or simulation of image data.

Neural Style Transfer is the main Style Transfer technique using deep learning [1]. The technique defines two distances, which express the difference in terms of content and style statistics between two images. Then, the key idea is to modify an input image such that it minimizes these distances with respect to the content and style images. However, computing the two distances requires using a pre-trained CNN on a large database.

Here, we want to leverage SinGAN's ability to learn the statistics in a single image, to perform style transfer. We hope that one single painting is enough to learn expressive statistics of a painter at different scales, that could be transferred to content images.

2.2. Method

As explained in the original SinGAN paper, injecting a particular image during the model evaluation process can help to retrieve specific elements of the injected image. We choose to leverage this property to perform style transfer.

Concretely, our method is the following: first, we train SinGAN on the style image, such that it learns the internal style statistics of the painting. Then, at inference time, we inject our chosen content image in the network and look the result that we obtain. As the injection can be done at every scale (except for the first one where the input is pure noise), we repeat the injection process and look for the best compromise between content and style preservation.

The sooner the content image is injected, the more the style appears (but the content is less likely to be preserved

by the pyramidal generation process). Conversely, if the content image is injected lately, it will undergo less operations and the content will be very preserved (but the style will less appear). An general illustration of the method can be found in figure S2.

2.3. Results

2.3.1 Training Variability

SinGAN does not learn the patch distribution as well on each artwork. This is particularly noticeable when drawing random samples using the model trained on the original image. In figures S3 and S4, the samples drawn from the model trained on the Warhol artwork look very much like the original - however, on the Kandinsky abstract painting with very complex small structures, the samples are immediately distinguishable.

Quantitatively speaking, we measure the difference in patch distribution between the samples and the artwork by evaluating the average Single Image Fréchet Inception Distance (SIFID) over 50 fake samples. The results are reported in table 1, and the corresponding artworks are shown in figure S5.

Artist (type)	Warhol (portrait)	Kandinsky (abstract)	Van Gogh (portrait)
SIFID	0.0431	0.2639	0.2167
Picasso (portrait)	Seurat (landscape)	Van Gogh (landscape)	Monet (landscape)
0.1156	0.0447	0.1373	0.0878

Table 1. SIFID between fake samples and original image

The average SIFID varies up to a factor 10 for different paintings. Hence, this patch learning difference on various images leads to differences in reliability when it comes to accurately transferring the style on the content image.

2.3.2 Portraits and Landscapes

First, we evaluate SinGAN's performance for Style Transfer on portraits. Our results can be found in figure S6. The resulting image succeeds in both preserving the global structure of the content image and injecting some stylistic traits of the painter. However, SinGAN struggles for preserving a plausible content portrait when injecting at early scales, so we have to inject it in late scales and miss some style elements. Moreover, the portraits do not contain enough style elements to be able to generalize to new colors and structures : for instance, it has difficulties to deal with the background.

Then, we choose to test its performances on landscapes paintings. Our idea is that the plausibility of landscapes is more robust to large structural changes: indeed, we are more prone to accept a deformation of a tree or a hill, rather than a deformation of an eye or a nose for instance. Also, small scale statistics such as textures and brushstroke styles are more important in this kind of paintings. Our results are reported in figure S7. As we can see, SinGAN succeeds in transfering the style at a small scale level (see the difference in the dots and brushstrokes across the paintings). However, some blurry artifacts remain.

2.3.3 Comparison with Neural Style Transfer

We decide to compare our method with the *Neural Style Transfer* (NST) technique to see if it is able to yield better or equivalent results: see figures S8 and S9. Our observation is that SinGAN and NST succeed at different levels: overall, NST leads to a better color transfer, and works with higher resolution images. However, the geometric style of the painter is poorly injected (see the Picasso for instance), and some color artifacts appear with high frequency variations (see the Van Gogh and Warhol paintings).

3. Application of SinGAN on PCG

Procedural Content Generation (PCG) refers to the use of algorithms to generate digital or game content, such as maps, levels or quests. PCG research is for instance motivated by the need to make games replayable or more recently, the need to procedurally generate learning environments for RL agents[10].

3.1. Related work

Here, we focus on level generation for the Mario AI Benchmark. A level is composed of ASCII tokens representing a type of block ('g' for Gomba, 'X' for Ground, etc.). It can be visualized as an image using existing sprites (See S10). The choice comes mainly from the popularity of the Mario AI Benchmark among academic AI and PCG researchers, meaning that this is probably the game on which the more literature is available.

This literature is methodologically very diverse, including approaches using agents, grammars, evolution, etc. and we refer the interested reader to [3]. Among these methods, Wave Function Collapse (WFC) is very common and generates new levels in the style of given examples by ensuring every local window of the output occurs somewhere in the input[4]. More recently, GANs combined with CMA-ES have also been used to create Mario levels as in [9].

3.2. Our SinGAN Method

Here, we propose a new method that exploits the ability of SinGAN to capture the statistics of the training image at different scales.

It consists in:

1. Train SinGAN on the image-level generated from the first Mario level of the Video-Game-Level-Corpus

(VGLC)[8]. Impose the number of SinGAN's scales to 8, as well as the final resolution (e.g. half of the original resolution).

2. Generate new image samples using noise injected at the first scale of SinGAN's pyramid.

3. Reconstruct ASCII-levels from the images, using template matching. The template matching method is performed in 3 steps: for each 8×8 patch in the generated image (the patch size depends on the resolution chosen at 2), first compute the Normalized Sum of Squared Difference (NSSD) score between the patch and each sprite (sprite for 'g', 'X', etc.). Then, multiply each token score by a prior depending on the number of times this token appears in the original level (for instance the *sky* token is much more frequent than the *pipe* one). Finally, choose the token which has minimum score.

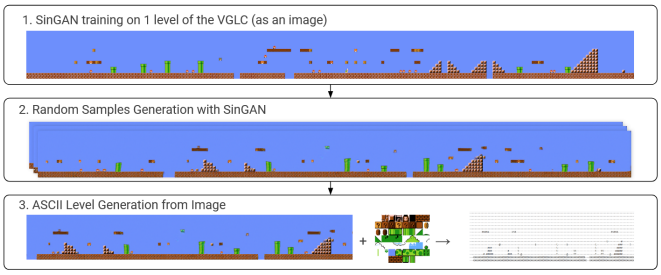

Figure 1. Our method to use SinGAN on level generation

3.3. Qualitative Results

We report several generated levels using our SinGAN method in S12. The method achieves to create very plausible levels, with non trivial elements at the right place, such as enemies or hidden blocks. It can be noted that some blocks are harder to reconstruct. For instance pipes are often reconstructed with width 1, and some gombas are flying. Therefore, the reconstruction step could be improved, for example by imposing some rules on the generation like the impossibility to create flying enemies.

3.4. Quantitative Results

One last method that is interested to mention here introduces a metric between the generated levels and the original level. Originally, it was also used as fitness in ETPKL to do evolutionary search, to find new coherent levels [5]. This metric is called the Tile Pattern KL (TPKL) divergence and reads $D_{KL}(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$, where P is the distribution of $n \times n$ pattern in the generated level and Q the one in the original level. This allows us to quantitatively compare different level generators by averaging the TPKL divergence on a high number of generated levels. The lower the distance is, the better the generator is, but the closer it stays to the original level. We report the TPKL divergence for our method, and compare it to the other methods mentioned in the Related Work. Our method obtains results very comparable to other PCG methods.

Method	GAN	WFC 2x2	WFC 3x3	WFC 4x4
TPKL 4x4	2.62	2.67	2.52	2.52
TPKL 3x3	1.63	1.58	1.61	1.67
TPKL 2x2	0.85	0.63	0.89	1.00

ETPKL 2x2	ETPKL 3x3	ETPKL 4x4	**SinGAN**
2.73	**1.91**	**1.53**	**1.93**
1.34	**0.57**	**0.90**	**1.14**
0.16	**0.16**	**0.47**	**0.55**

Table 2. TPKL divergence for $n \times n$ window, averaged on 50 levels. The divergence is computed between the generated levels and the original Mario level for different PCG methods. The results of other methods are taken from [5].

3.5. Other PCG Applications

Here, we focused on only one level. Further work would consist in trying the method for other games and levels available in the VGLC, and see if our results generalize to these levels. Another interesting idea would be to use SinGAN to generate new maps for games, books or movies, using a single instance for training to get many new variations of plausible results. An example is given in S11.

4. Conclusion

Our main contributions for this project are the extension and evaluation of SinGAN's method for two new applications: the first one is Style Transfer, and the second one is Procedural Content Generation. For both methods, we provide qualitative and quantitative evaluations, as well as comparison with other methods.

As one can notice, the two applications seem to give satisfying and plausible results. Style Transfer led to interesting artworks in a different way than NST, and PCG led to plausible game levels as well. Nonetheless, from a practical point of view, SinGAN's method seems a bit cumbersome, since it has to be trained on each specific example. Our experiments showed that SinGAN's training takes around 2 hours on an single 256×256 image on a Google Colab GPU. One would maybe prefer more end-to-end method, which would be trained once and for all, and that could be extensively used at inference time.

Be that as it may, learning the statistics of an image on a multi-scale level still is an interesting idea that could be more deeply explored. One of our ideas was to think about possible extensions on small batches of images (e.g. a batch of paintings from a painter, or a batch of Mario levels) rather than one single image, for better scalability and robustness.

References

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 1

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014. 1

[3] Mark Hendrikx, Sebastiaan Meijer, Joeri Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 9, 02 2013. 2

[4] Isaac Karth and Adam M. Smith. Wave function collapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, New York, NY, USA, 2017. Association for Computing Machinery. 2

[5] Simon M. Lucas and Vanessa Volz. Tile pattern kl-divergence for analysing and evolving game levels. *Proceedings of the Genetic and Evolutionary Computation Conference*, Jul 2019. 1, 3

[6] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[7] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Computer Vision (ICCV), IEEE International Conference on*, 2019. 1

[8] Adam James Summerville, Sam Snodgrass, Michael Mateas, and Santiago Onta n'on Villar. The vglc: The video game level corpus. *Proceedings of the 7th Workshop on Procedural Content Generation*, 2016. 3

[9] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam Smith, and Sebastian Risi. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 221–228, New York, NY, USA, 2018. Association for Computing Machinery. 2

[10] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, NY, USA, 2019. Association for Computing Machinery. 2

Supplementary Materials

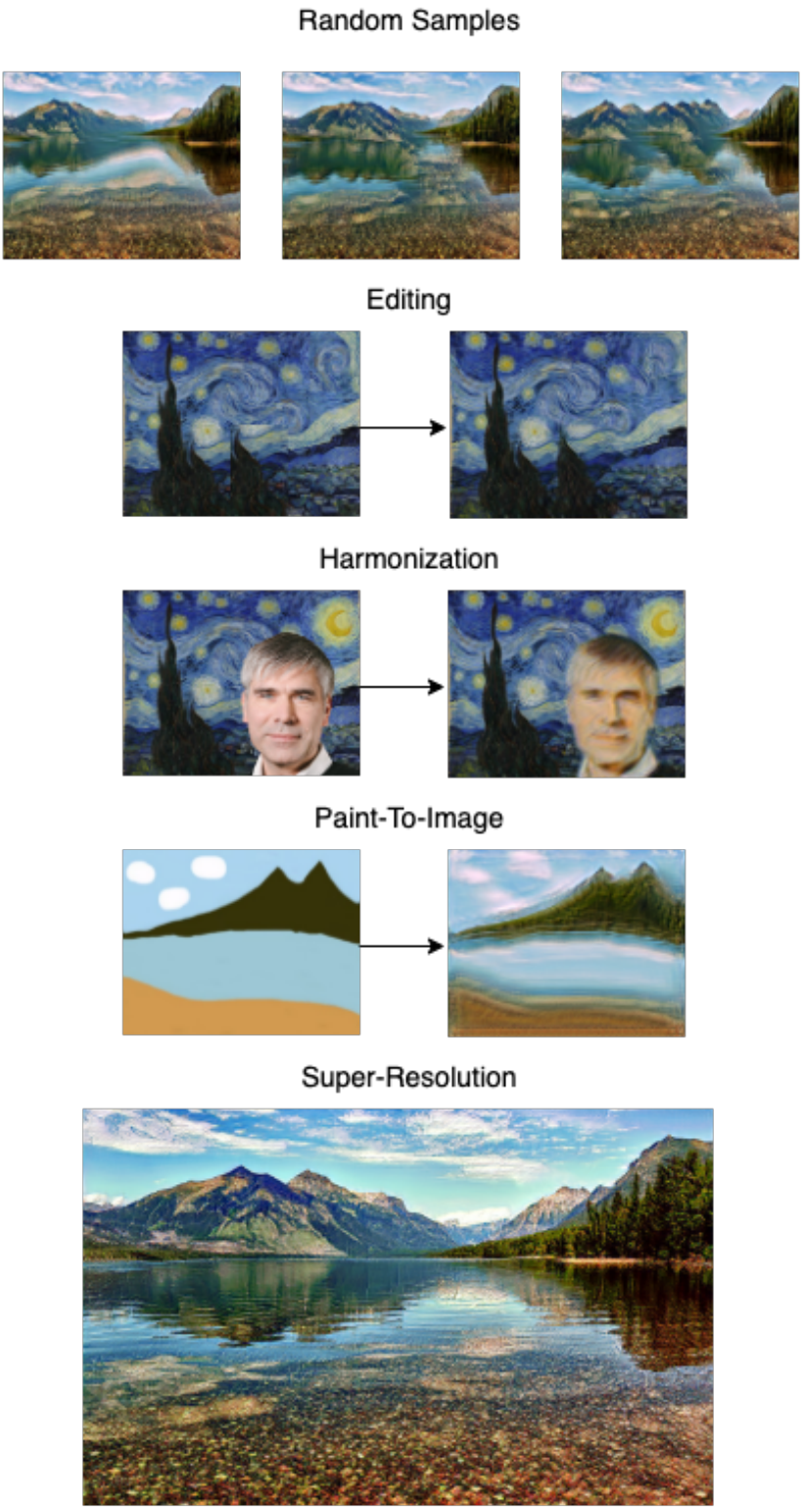

Figure S1. SinGAN results on various image manipulation tasks. An example of an animation result is available in our slides.

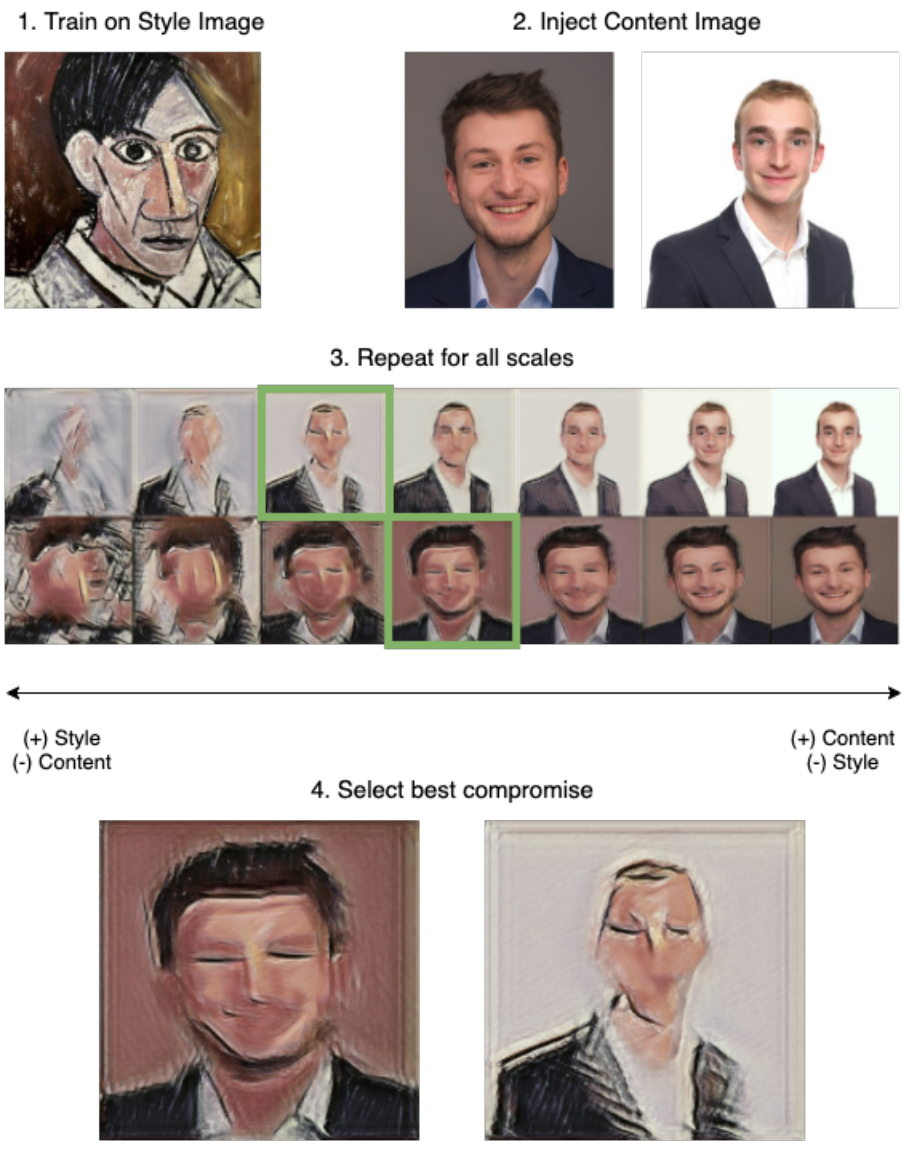

Figure S2. Style Transfer using SinGAN: Method

Figure S3. An artwork with low variability on the fake samples

Figure S4. An artwork with high variability on the fake samples

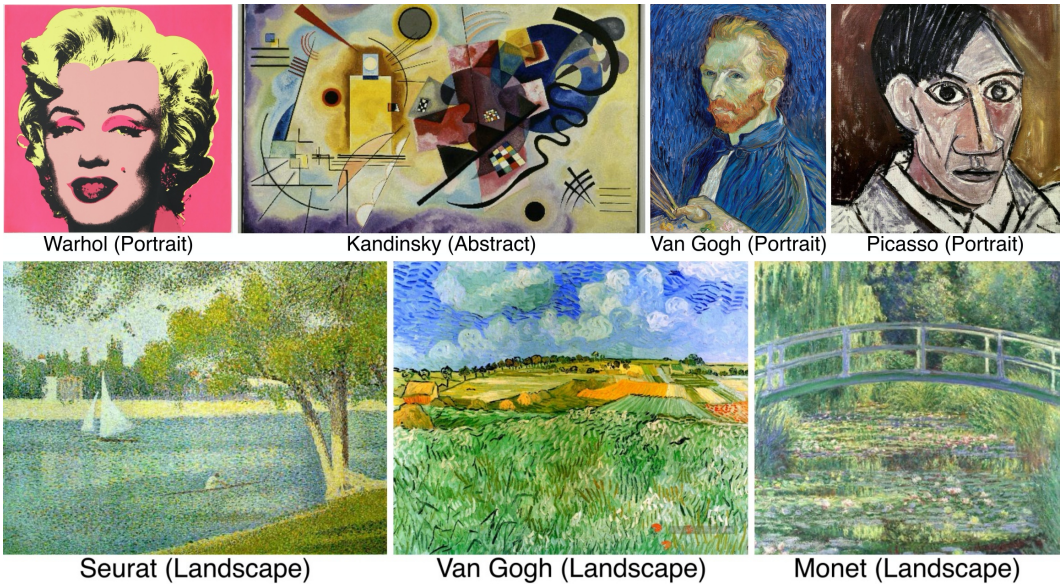

Figure S5. Artworks used for the variability study

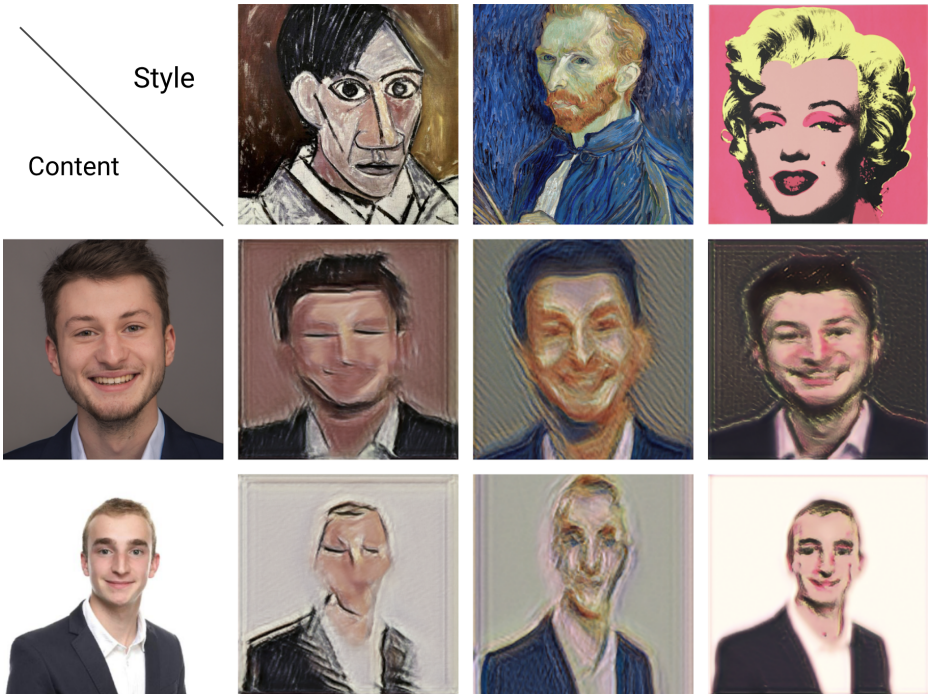

Figure S6. Style Transfer on portraits with SinGAN

Figure S7. Style Transfer on landscapes with SinGAN

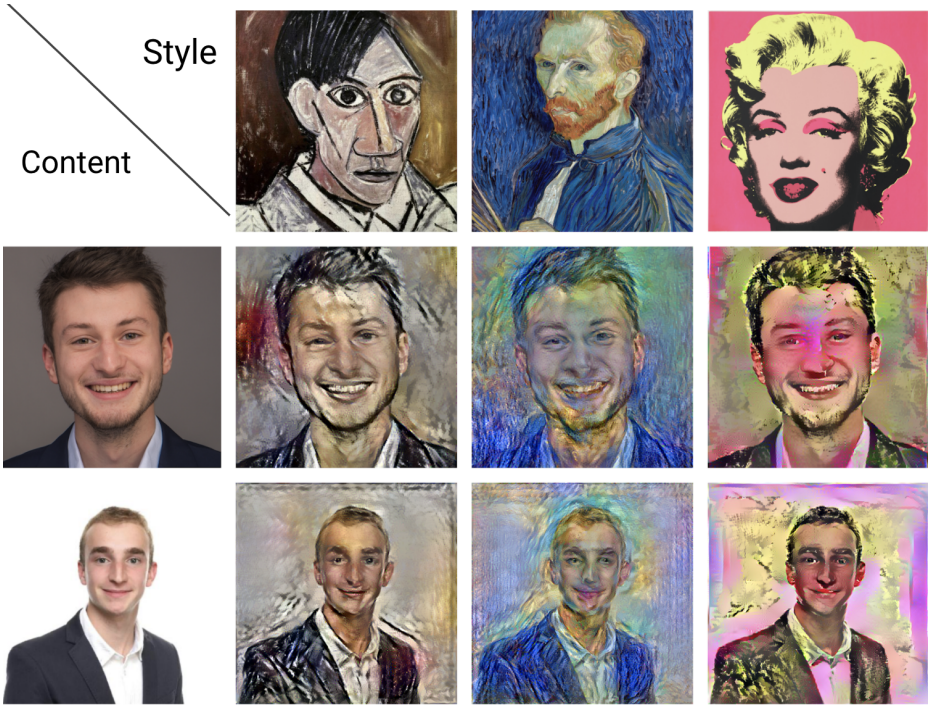

Figure S8. Style Transfer on portraits with NST

Figure S9. Style Transfer on landscapes with NST

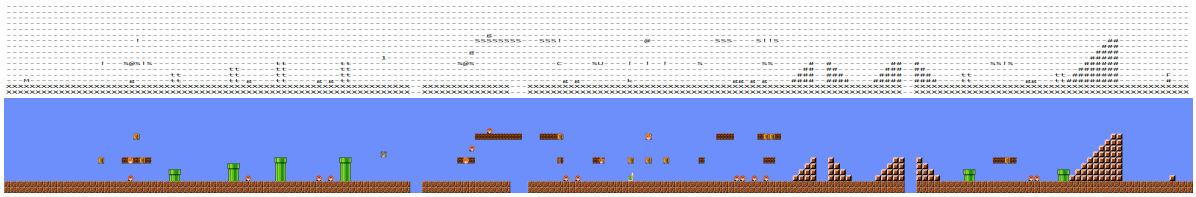

Figure S10. Level 1 of the Mario VGLC (up: ASCII level, down: image generated from sprites)

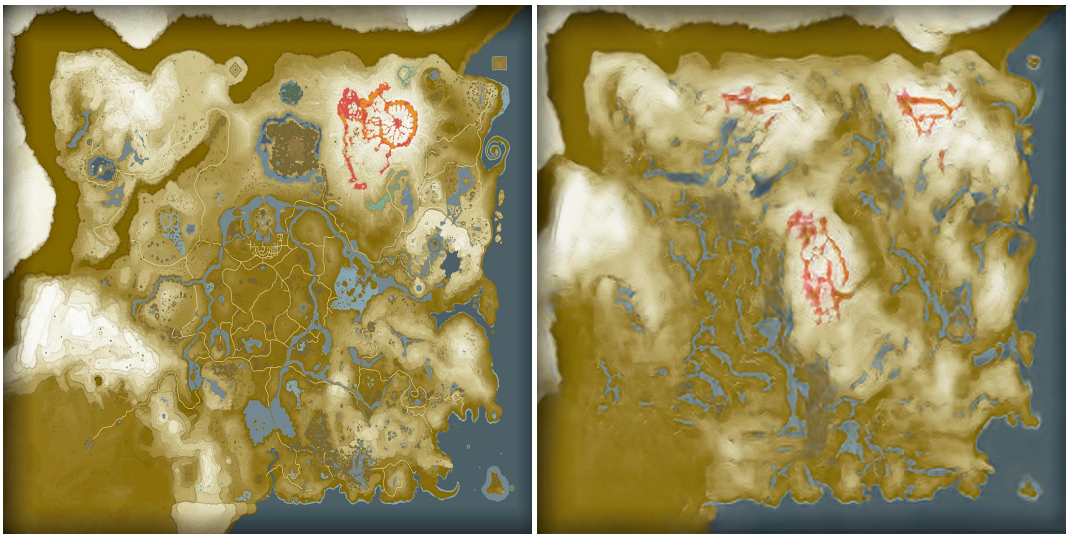

Figure S11. Map generation example using the map from The Legend of Zelda: Breath of the Wild. Left: original, right: fake sample

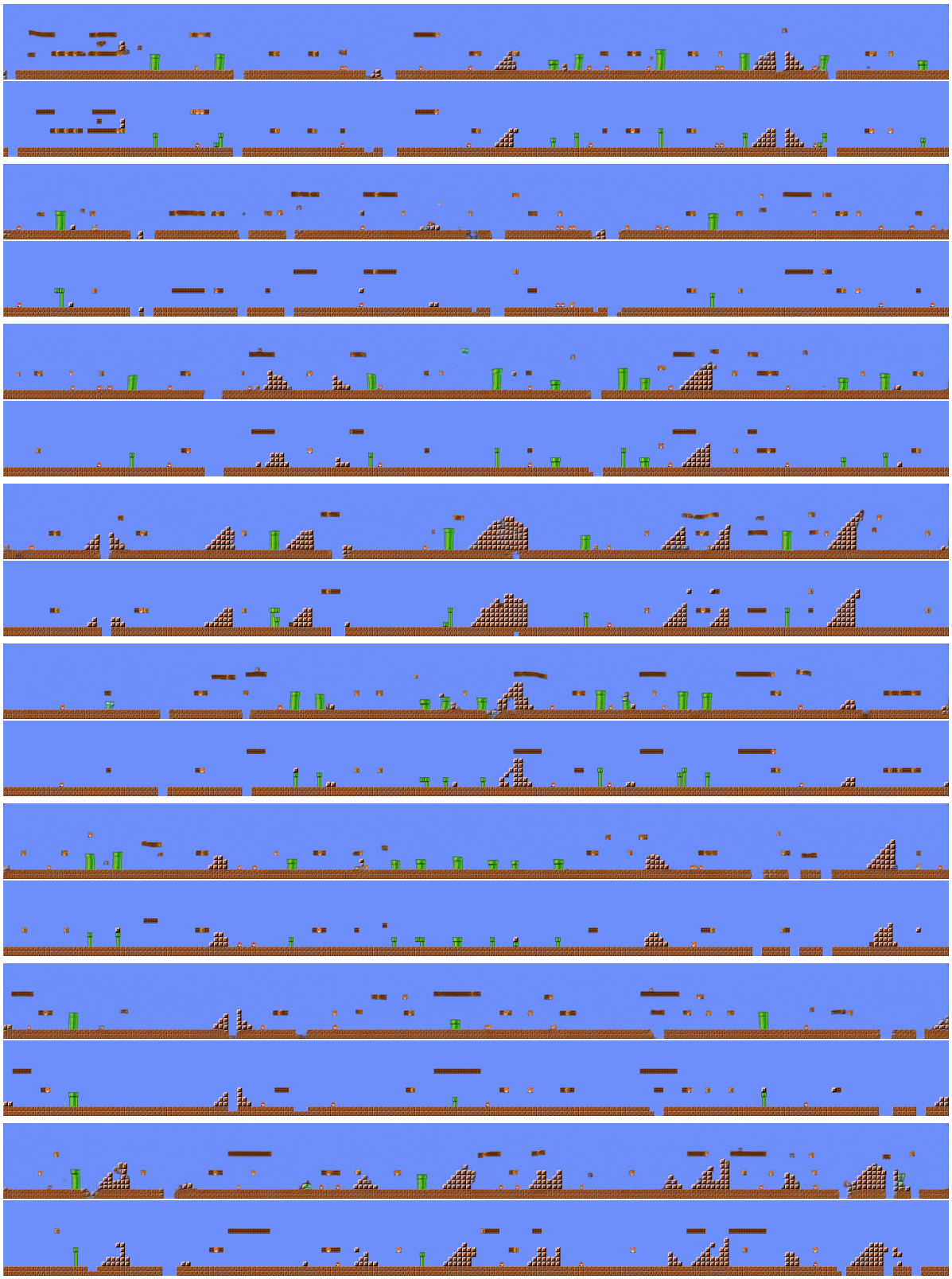

Figure S12. Levels generated with our SinGAN method at the 2 steps of the generation. Up: image generated by SinGAN, down: reconstructed level with template matching